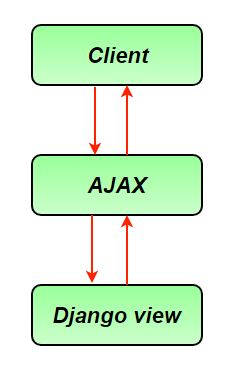
Django Ajax

how to carry out an ajax request in the Django web framework



**Implementation:**

**1. Initiate the Django Project**

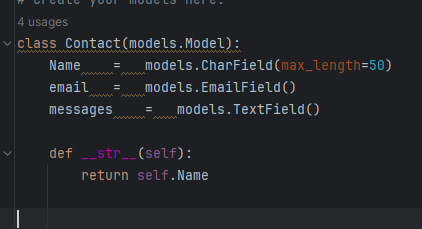
* Ensure Django is installed (pip install django) and create a new project (django-admin startproject projectname).
* Create a new app (python manage.py startapp appname) where you'll handle the AJAX form submission.

django-admin startproject AjaxProject .

python manage.py startapp AjaxApp

**2. Define Your Model:**

* Define the model for the data you want to collect with the form in models.py.



**3.Make Migration and migrate step:**

python manage.py makemigrations

python manage.py migrate

**4.Add bootstrap link**

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

**5.Add JQuery Link**

<script

src="https://code.jquery.com/jquery-3.5.0.min.js"

integrity="sha256-xNzN2a4ltkB44Mc/Jz3pT4iU1cmeR0FkXs4pru/JxaQ="

crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"

crossorigin="anonymous"></script>

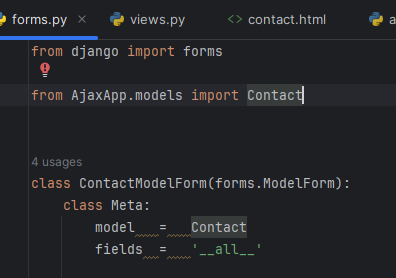
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"

integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"

crossorigin="anonymous"></script>

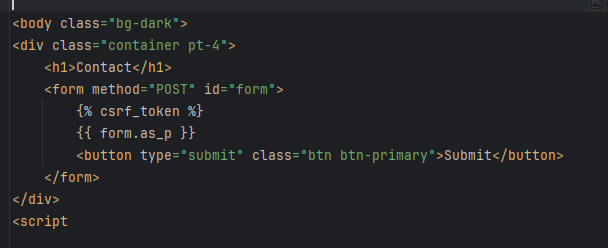
**6. Create a Form:**

* Create a Django ModelForm in forms.py that corresponds to your model.

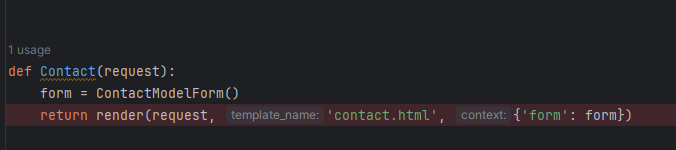


**7. Design Your Template:**

* Create an HTML template (e.g., form.html) that includes your form.
* Make sure to load jQuery library and CSRF token in the template.

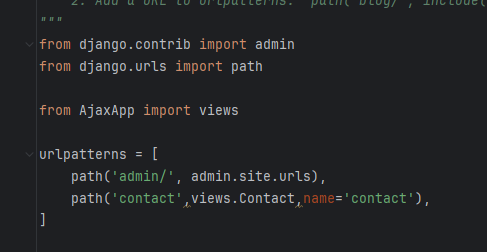
**8. Create a View:**

* Implement a view function in views.py to handle form submission.

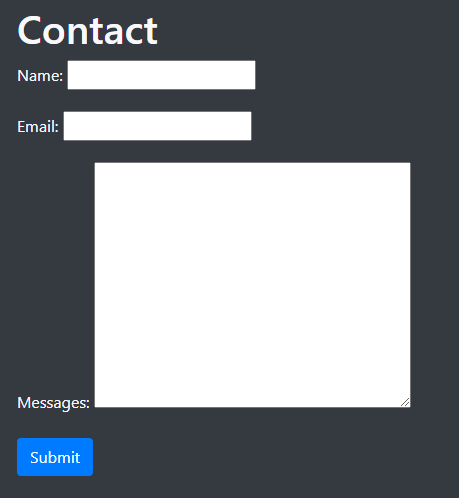


**9. Configure URLs:**

* Define a URL pattern in urls.py for your form submission endpoint.

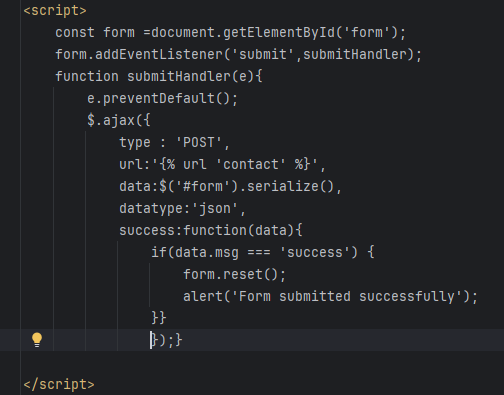


**10.python manage.py runserver**



**11. Implement AJAX Functionality:**

* Write JavaScript code in the template to handle form submission via AJAX.
* Use jQuery to select the form and attach a submit event listener.
* Prevent the default form submission behaviour.
* Serialize form data and send it to the server using AJAX POST request.
* Handle the server response, e.g., display success or error messages.



Explanation:

* 1)**const form = document.getElementById('form');:**
* This line selects the form element with the ID 'form' from the HTML document and stores it in the variable form.
* 2)**form.addEventListener('submit', submitHandle**r);:
* This line adds an event listener to the form. When the form is submitted, it calls the submitHandler function.
* 3)**function submitHandler(e) { ... }:**
* This is the submitHandler function, which is triggered when the form is submitted.
* 4)e.**preventDefault**();:
* This line prevents the default form submission behavior, which would cause the page to reload.
* 5)$.ajax({ ... }):
* This is a jQuery function for making AJAX requests. It sends a POST request to the server.
* 6)**type: 'POST':**
* Specifies that the request method is POST.
* **7)url: '{% url 'contact' %}'**:
* This is a Django template tag ({% ... %}) that dynamically generates the URL for the form submission endpoint named 'contact'.
* **8)data: $('#form').serialize():**
* Serializes the form data into a URL-encoded string that can be sent to the server. $('#form') selects the form element using jQuery.
* **9)dataType: 'json':**
* Specifies that the expected data type of the response from the server is JSON.
* **10)success: function(data) { ... }:**
* Defines a callback function to handle the successful response from the server. The data parameter contains the response data.
* **11)if (data.msg === 'success') { ... }:**
* Checks if the response data contains a property msg with the value 'success'. If so, it means the form submission was successful.
* **12)form.reset();:**
* Resets the form fields, clearing any user-entered data.
* **13)alert('Form submitted successfully');:**
* Displays an alert message informing the user that the form was submitted successfully.

